

### **Amendments to the Claims**

This listing of claims will replace all prior versions, and listings, of claims in the application:

#### **Listing of Claims:**

1. (Previously presented) A method for restoring a persistently stored object of an object oriented environment established on a computer system having a volatile memory and a persistent storage, the method comprising steps of:
  - retrieving from said persistent storage a first list of first references to segments stored in said persistent storage;
  - retrieving all segments referenced in the first list and storing the segments in said volatile memory;
  - saving in said first list a difference between an old memory address at which the segment used to reside in the volatile memory and a new memory address at which said segment is currently stored;
  - retrieving from said persistent storage a second list of second references to blocks, wherein the blocks are portions of segments and whereby at least one of said blocks contains an object description;
  - determining, from the second list, which segment contains the block where the object description of the persistently stored object is located;
  - having determined the segment in which the object description of the persistently stored object resides, retrieving the objection description;
  - creating a first object in said volatile memory using said object description retrieved from said segment; and
  - saving a new address of said first object in said second list in volatile memory;
  - initializing said first object with values taken from said object description;
  - determining the address of a second object description of a second object referenced in said first object; and
  - setting the address of said second object description as a reference in the object

description of the first object.

2. (Previously presented) The method according to claim 1, whereby said first list and/or said second list are ordered lists.

3. (Previously presented) The method according to claim 1, whereby said first list and/or said second list are organized as a B-tree.

4. (Currently amended) The method according to claim 2, whereby elements of said first ~~ordered~~ list are indexed by said first references.

5. (Previously presented) The method according to claim 1, whereby each of said first references corresponds to the old memory address at which the segment used to reside in the volatile memory.

6. (Currently amended) The method according to claim 2, whereby elements of said second ~~ordered~~ list are indexed by said second references.

7. (Previously presented) The method according to claim 1, whereby each of said second references corresponds to an old memory address at which the block used to reside in said volatile memory.

8. (Previously presented) The method according to claim 1, whereby said first and second object descriptions are formed by a collection of values owned by an object for variables belonging to its class.

9. (Previously presented) The method according to claim 1, whereby for each value in said first and second object descriptions of variables having a variable length, the method further comprising steps of:

allocating a number of blocks, such that an actual value of the variable having a variable length can be kept variable;

creating a linked list of said number of blocks;

saving said value into said number of blocks; and

storing a reference to the head of the linked list in said first and second object descriptions.

10. (Previously presented) The method according to claim 2, whereby determining the segment that contains said block referenced by a particular element of said second list comprises the step of searching in said first ordered list for the segment that contains said portion of said segment referenced by said element.

11. (Previously presented) The method according to claim 1, whereby determining the segment that contains said block referenced by a particular element of said second list further comprises the step of calculating the new address by adding the reference to said block, that corresponds to the old memory address, and said difference between said old memory address and said new memory address.

12. (Previously presented) The method according to claim 1, whereby determining new addresses of objects referenced by a newly created object comprises the step of searching in said second list for the element that contains the new address of the referenced object, that is referenced by the old address of the respective object description.

13. (Currently amended) The method according to claim 1, whereby said first list and/or said second list is a linked list, and

whereby for all references to heads of the linked list the method further comprising the steps of:

- reading all blocks of said linked list;
- allocating memory to store the value of the variable retrieved from the linked list; and
- storing the value in said allocated memory.

14. (Previously presented) A method for persistently storing objects of an object-oriented environment established on a computer system having a volatile memory and a persistent storage, the method comprising steps of:

- allocating a plurality of segments in said volatile memory;
- creating a first list of first references to said segments;
- creating a second list of second references to blocks, wherein the blocks are portions of said segments;
- allocating a block of one of said segments;
- creating a first object description for a first object, wherein the first object description comprises values of variables belonging to the first object's class;
- storing the first object description into said allocated block;
- saving a new address of said first object description in the second list, wherein the new address is a memory address of said allocated block in the volatile memory where the first object description is stored;
- adding reference to said first object description to the second list;
- determining the address of a second object description of a second object referenced in said first object description;
- setting the address of said second object description as a reference in the created first object description;
- storing said second list on said persistent storage;
- storing the plurality of segments referenced by said first list on said persistent storage;

and

storing said first list on said persistent storage.

15. (Previously presented) The method according to claim 14, whereby said first list and/or said second list are ordered lists.

16. (Previously presented) The method according to claim 14, whereby said first list or said second list are organized as a B-tree.

17. (Currently amended) The method according to claim 16, whereby elements of said first ~~ordered~~ list are indexed by said first references.

18. (Previously presented) The method according to claim 14, whereby each of said first references corresponds to the current memory address at which the segment resides in the volatile memory.

19. (Currently amended) The method according to claim 15, whereby elements of said second ~~ordered~~ list are indexed by said second references.

20. (Previously presented) The method according to claim 14, whereby each of said second references corresponds to the current memory address at which the block resides in said volatile memory.

21. (Previously presented) The method according to claim 19, whereby determining an address of an object description of another object referenced in said first object comprises the step of searching in said second ordered list for the element that contains the address of the object description of the another object.

22. (Previously presented) The method according to claim 21, whereby determining the

address of the object description of another object referenced in said first object further comprises a step of using a reference to the another object description provided by each object in said second ordered list.

23. (Currently amended) The method according to claim 14, whereby for each value of the variables having a variable length the method further comprises steps of:

- allocating a number of portions of one of said segments of memory such that an actual value of the variable length can be kept variable;
- creating a linked list of said number of portions;
- saving the values of the variables into said number of portions; and
- storing a reference to ~~the~~ a head of the linked list in said object description.

24. (Previously presented) A computer program product stored on a computer usable medium, comprising computer readable program instructions for:

- allocating a plurality of segments in volatile memory;
- creating a first list of first references to said segments;
- creating a second list of second- references to blocks, wherein the blocks are portions of said segments;
- allocating a block of one of said segments,
- creating a first object description of a first object, wherein the first object description comprises values of variables belonging to the first object's class;
- storing the first object description into said allocated block;
- saving a new address of said first object description in the second list, wherein the new address is a memory address of said allocated block in the volatile memory where the first object description is stored;
- adding a reference to the first object description to the second list;
- determining the address of a second object description of a second object referenced in said first object description;

setting the address of said second object description as a reference in the created first object description;  
storing said second list on persistent storage;  
storing the plurality of segments referenced by said first list on said persistent storage;  
and  
storing said first list on said persistent storage.